

ARSITEKTUR SOFTWARE AS A SERVICE (SAAS) PADA LAYANAN RUMAH SAKIT

Toni Kusnandar

Magister Informatika, Sekolah Teknik Elektro dan Informatika

Institut Teknologi Bandung, Jl Ganesha 10 Bandung

toni.kusnandar@gmail.com

Abstract

Cloud Computing is a trend that includes several categories of services, all offered in the form of an Internet service with a model pay for what is used by the user (pay as you go). Software as a Service (SaaS) is one in this category, in addition to Platform as a Service (PaaS) and Infrastructure as a Service (IaaS). SaaS is the most mature category of Cloud Service, as evolved from the model application service-provider. Hospital as an institution that provides service every day of non-stop for 24 hours, began requiring the application of Cloud Computing to save operating costs of data processing, improve agility, but still can involve a lot of users. Therefore we need an appropriate architecture to meet the expectations of the Hospital. SaaS architecture in hospital services will contribute to the Data Architecture, Application Architecture and Technical Architecture will even contribute also to the Organizational Architecture.

Keywords: Cloud Computing, Software as a Service

Abstrak

Cloud Computing merupakan tren yang mencakup beberapa kategori layanan, semuanya ditawarkan dalam bentuk layanan Internet dengan model membayar terhadap apa yang digunakan oleh pengguna (pay as you go). Software as a Service (SaaS) adalah salah satu dalam kategori ini, selain Platform as a Service (PaaS) dan Infrastructure as a Service (IaaS). SaaS adalah kategori yang paling matang dari Cloud Service, karena berevolusi dari model application service-provider. Rumah Sakit sebagai institusi yang memberikan layanan setiap hari non-stop selama 24 jam, mulai membutuhkan penerapan Cloud Computing untuk menghemat biaya operasional pengolahan data, meningkatkan agility, tetapi tetap dapat melibatkan banyak pengguna. Oleh karena itu diperlukan sebuah arsitektur yang tepat untuk memenuhi harapan Rumah Sakit tersebut. Arsitektur SaaS pada layanan RS akan memberikan kontribusi pada Data Architecture, Application Architecture dan Technical Architecture bahkan akan berkontribusi juga pada Organizational Architecture.

Kata kunci : *Cloud Computing, Software as a Service*

1. PENDAHULUAN

Cloud Computing merupakan tren yang mencakup beberapa kategori layanan, semuanya ditawarkan dalam bentuk layanan Internet dengan model membayar terhadap apa yang digunakan oleh pengguna (*pay as you go*). *Software as a Service* (SaaS) adalah salah satu dalam kategori ini, selain *Platform as a Service* (PaaS) dan *Infrastructure as a Service* (IaaS). SaaS adalah kategori yang paling matang dari *Cloud Service*, karena berevolusi dari model *application service-provider*.

Penggunaan SaaS telah menghasilkan berbagai manfaat seperti penghematan biaya, ketangkasan (*agility*) yang lebih baik, dan mempercepat penyampaian layanan kepada pengguna, serta peningkatan fleksibilitas dalam berbagai skala untuk mendukung lebih banyak pengguna yang diperlukan. Hal ini juga menyediakan tempat untuk bereksperimen dengan kemampuan baru.

Rumah Sakit sebagai institusi yang memberikan layanan setiap hari non-stop selama 24 jam, mulai membutuhkan penerapan *Cloud Computing* untuk menghemat biaya operasional pengolahan data, meningkatkan *agility*, tetapi tetap dapat melibatkan banyak pengguna. Oleh karena itu diperlukan sebuah Arsitektur yang tepat untuk memenuhi harapan Rumah Sakit tersebut.

2. SISTEM INFORMASI

Sebuah sistem informasi dapat didefinisikan sebagai subsistem sosio-teknis dari suatu organisasi, yang terdiri dari seluruh komponen pengolahan informasi serta aktor manusia atau teknis yang terkait dalam peranan mereka dalam mengolah informasi (Winter). Komponen khas dari sistem informasi adalah:

- fungsi perusahaan yang didukung;

- proses bisnis yang sedang berlangsung;
- komponen aplikasi yang mendukung fungsi perusahaan;
- komponen aplikasi yang dijalankan pada sistem pengolahan data.

2.1. Sistem Informasi Rumah Sakit

Sistem informasi kesehatan (HIS) berhubungan dengan pengolahan data, informasi, dan pengetahuan dalam lingkungan pelayanan kesehatan. Khususnya mengenai penyakit kronis, menjadi lebih penting untuk mengatur perawatan kesehatan dengan cara pasien-sentris, sehingga semua peserta atau lembaga rawat jalan bekerja sama dengan sangat erat. Hal ini disebut sebagai perawatan terpadu.

Dalam perawatan terpadu memberikan informasi yang relevan tidak hanya dalam sebuah institusi tunggal, tetapi dimanapun dan kapan pun ia dibutuhkan. Hal ini termasuk praktek medis, pusat rehabilitasi, pusat perawatan, dan bahkan rumah pasien itu sendiri. Oleh karena itu kita membedakan sistem informasi kesehatan kelembagaan dan transinstitutional. Sistem informasi rumah sakit, yang merupakan kasus yang paling kompleks dapat berupa sistem informasi kelembagaan, dan sistem informasi transinstitutional.

3. SOFTWARE AS A SERVICE

3.1. Cloud Computing

Peter Mell dan Tim Grance dari Institut Nasional Ilmu Pengetahuan dan Teknologi (NIST) telah mengembangkan sekumpulan definisi sekitar *Cloud Computing* (Winkler). Dalam tulisan itu, mereka menyatakan bahwa *Cloud Computing* adalah paradigma yang masih berkembang, dan mereka menggambarkan *Cloud Computing* sebagai:

... model untuk memudahkan, berdasarkan permintaan akses jaringan secara bersama dalam sumber daya komputasi yang telah dikonfigurasi (misalnya, jaringan, server,

penyimpanan, aplikasi, dan jasa) yang dapat dengan cepat ditetapkan dan dirilis dengan upaya manajemen atau interaksi penyedia layanan yang minimal. Model *cloud* meningkatkan ketersediaan dan terdiri dari lima karakteristik penting, tiga model layanan, dan empat model penyebaran (Mell & Grance, 2009).

Mell dan Grance (Mell P, Grance, T. *The NIST Definition of Cloud Computing Version 15, 10-7-09, National Institute of Standards and Technology, Information Technology Laboratory*) mengidentifikasi lima karakteristik penting sebagai berikut (Winter):

- **On-Demand Self-Service.** Seorang konsumen secara sepihak dapat menentukan kemampuan komputasinya, seperti waktu server dan penyimpanan jaringan, sesuai kebutuhan secara otomatis tanpa memerlukan interaksi manusia dengan masing-masing penyedia layanan.
- **Broad Network Access.** Kapabilitas yang tersedia melalui jaringan dan diakses melalui mekanisme standar yang digunakan dengan mempromosikan platform heterogen baik thin client maupun thick client (misalnya, ponsel, laptop, dan PDA).
- **Resource Pooling.** Sumber daya komputasi penyedia dikumpulkan untuk melayani banyak konsumen menggunakan model *multi-tenant*, dengan sumber daya fisik dan virtual yang berbeda yang ditetapkan secara dinamis dan dipindahkan sesuai dengan permintaan konsumen. Ada perasaan kebebasan lokasi bahwa pelanggan umumnya tidak memiliki kontrol atau pengetahuan atas lokasi yang tepat dari sumber daya yang disediakan tetapi mungkin dapat menentukan lokasinya pada tingkat abstraksi yang lebih tinggi (misalnya, negara, negara bagian, atau pusat data). Contoh sumber daya termasuk penyimpanan, pemrosesan, memori,

bandwidth jaringan, dan mesin virtual.

- **Rapid Elasticity.** Kemampuan yang dapat dengan cepat dan elastis ditetapkan, dalam beberapa kasus akan secara otomatis, dengan cepat dikeluarkan dan dengan cepat dilepaskan untuk meningkatkan kecepatan masuk. Untuk konsumen, kapabilitas tersedia bagi pengadaan sering muncul menjadi tidak terbatas dan dapat dibeli dalam jumlah berapapun setiap saat.
- **Measured Service.** Sistem *Cloud* secara otomatis mengontrol dan mengoptimalkan penggunaan sumber daya dengan memanfaatkan kemampuan pengukuran di beberapa tingkat abstraksi yang sesuai dengan jenis layanan (misalnya, penyimpanan, pemrosesan, bandwidth, dan account pengguna aktif). Penggunaan sumber daya dapat dipantau, dikendalikan, dan dilaporkan dengan memberikan transparansi bagi penyedia dan konsumen dari layanan yang dimanfaatkan.

3.2. Definisi SaaS

Software as a Service (SaaS) berbagi peranan penting baik model bisnis dan model pengiriman aplikasi (Progress Software Corporation, White Paper: SaaS architecture). SaaS memungkinkan pelanggan untuk dapat memanfaatkan aplikasi berbasis *pay-as-you-go* dan menghilangkan kebutuhan untuk melakukan instalasi serta menjalankan aplikasi pada perangkat keras sendiri. Pelanggan umumnya mengakses aplikasi melalui browser Web atau *thin client* melalui Internet. SaaS umumnya berbasis langganan dan semua dukungan yang berkelanjutan, pemeliharaan, dan upgrade disediakan oleh perangkat lunak milik vendor sebagai bagian dari layanan tersebut. Kemampuan kustomisasi aplikasi, jika tersedia, umumnya diberikan kepada semua pelanggan secara konsisten. Dari perspektif vendor perangkat lunak,

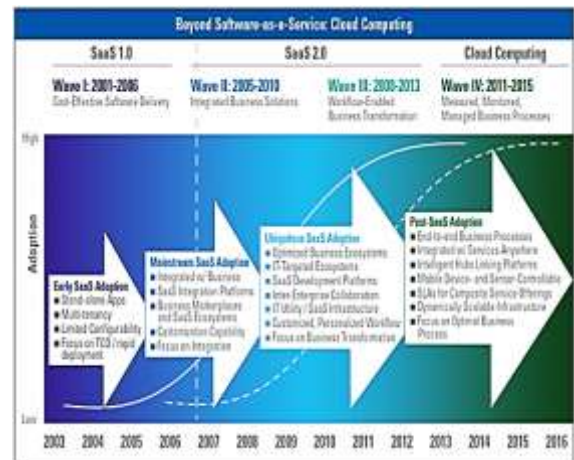
model SaaS menyediakan perlindungan yang lebih kuat terhadap hak kekayaan intelektualnya, lingkungan pengendalian operasional dalam menjalankan perangkat lunak, dan aliran pendapatan umumnya diperoleh dari biaya layanan berlangganan. Vendor perangkat lunak memiliki berbagai kemampuan dan aplikasi dapat datang dalam berbagai gaya, tapi aplikasi SaaS paling sering mendukung pelanggan yang unik yang menggunakan satu contoh dari aplikasi yang juga dikenal sebagai *multi-tenancy*.

3.3. Evolusi Model SaaS

Dalam Laporan Strategis tahun 2007, Saugatuck Teknologi mendefinisikan konsep evolusi pasar SaaS “Three Waves”, dan mereka telah memperluas hal ini dengan memasukkan Wave keempat sehingga menjadi Teori “Four Wave”. Konsep ini memberikan pendekatan yang berguna untuk memahami model *emerging* dan jatuh tempo SaaS dan dapat membantu vendor perangkat lunak yang ingin pindah dari yang berbasis ASP (penyedia layanan aplikasi) atau aplikasi *client-server* kepada SaaS. Gambar 1 menunjukkan diagram dari empat gelombang dari *paper* Saugatuck terbaru.

Gelombang Pertama, vendor SaaS memberikan penawaran yang sangat sederhana biasanya melibatkan aplikasi yang berdiri sendiri (umumnya baik entry-level atau fungsi khusus) dan fokusnya adalah pada penghematan biaya karena kemudahan penyebaran, pemeliharaan dan penggunaan. Sebuah artikel terbaru dari Software CEO mencatat IDC “Black Book Worldwide” menemukan bahwa hanya 20% dari anggaran TI dialokasikan untuk perangkat lunak sementara sisanya untuk sumber daya manusia, layanan dan hardware yang dibutuhkan untuk menjalankan software ini. Pengguna akhir (pada tingkat perusahaan) merasa terdapat ketidaksesuaian antara biaya yang dikeluarkan dengan nilai yang diperoleh, sehingga gelombang pertama

ditempatkan sebagai nilai proposisi dasar.



Gambar 1. Saugatuck SaaS Wave Evolution Model Source: Saugatuck Technology Inc.

Gelombang ke Dua, bagian akhir dari yang merupakan tempat kita menemukan diri kita hari ini, adalah penerapan SaaS dan solusi yang lebih canggih. Di sini kita melihat munculnya kemampuan kustomisasi lebih lanjut dan fokus utamanya pada integrasi. SaaS mengambil peran lebih besar dalam ekosistem perusahaan secara keseluruhan, kemampuannya untuk mengintegrasikan solusi ini sebagai bagian dari aliran data perusahaan secara keseluruhan menjadi sangat penting.

Gelombang ke Tiga, SaaS berkembang, dewasa dan keuntungan lebih banyak diterima secara luas ke titik di mana ia menjadi bagian penting dari strategi perangkat lunak perusahaan secara keseluruhan. Akibatnya, menjadi fasilitator transformasi bisnis. Integrasi antara layanan bisnis SaaS dan layanan *on-premise*, serta berbagi data antar perusahaan, adalah fitur utama dari gelombang ini. Sebuah contoh yang sangat baik adalah integrasi aplikasi sumber daya manusia bagi perusahaan seperti *Oracle PeopleSoft* dengan solusi CRM SaaS. Hal ini mengandalkan solusi

SaaS terintegrasi dengan struktur sumber data perusahaan yang ada yang memungkinkan transformasi bisnis itu sendiri, sehingga proses yang ada lebih efisien dan hemat biaya dan dalam beberapa kasus menciptakan proses bisnis baru yang sebelumnya tidak layak.

Gelombang ke Empat, membahas munculnya paradigma *cloud computing* dan menduga bahwa memanfaatkan lingkungan baru ini "adalah perkembangan alami untuk SaaS, konsep utilitas TI, dan proses bisnis *outsourcing* dan transformasi." *Cloud computing* menawarkan infrastruktur *on-demand* dan dikombinasikan dengan perangkat lunak *on-demand* (SaaS) dapat memberikan solusi yang lebih fleksibel untuk digunakan dan dikelola, melayani percepatan transformasi bisnis yang menandai gelombang ketiga.

4. ARSITEKTUR

4.1. Arsitektur Enterprise

Enterprise Architecture (EA) adalah sebuah metode dan prinsip pengorganisasian yang menyelaraskan tujuan bisnis dan strategi fungsional dengan strategi teknologi informasi dan rencana pelaksanaan (Oracle Technology Network). Ada berbagai alternatif untuk mengembangkan EA termasuk *Framework Zachman*, *The Open Group Architecture Framework (TOGAF)*, *Enterprise Architecture Federal* dan *Metodologi Gartner*.

Masing-masing metode memiliki kekuatan dan kelemahan. Dalam praktek mengembangkan pendekatan EA yang paling umum sering kali merupakan perpaduan antara metode yang telah ada. EA dapat digunakan untuk menggambarkan metode untuk merancang sistem informasi rumah sakit dalam mendefinisikan sekumpulan blok bangunan, dan menunjukkan bagaimana komponen pembentuk cocok satu sama lain serta bagaimana komunikasi antara

blok bangunan dapat dicapai. Sejak dikembangkan pada tahun 1984 pendekatan EA telah diterapkan oleh banyak perusahaan, pemerintah dan lembaga lain di seluruh dunia dalam rangka meningkatkan proses bisnis mereka, misalnya Departemen Pertahanan AS, *Massachusetts Institute of Technology*, perusahaan-perusahaan komersial seperti BP (*British Petroleum*), Intel dan Volkswagen. Organisasi global seperti Bank Dunia sedang melihat EA sebagai enabler untuk reformasi yang luas di sektor publik (Stansfield, Orobato, Lubinski, Uggowitz, & Mwanyika).

Langkah penting dalam mengembangkan arsitektur enterprise adalah untuk menetapkan ruang lingkup awal. Sebuah ruang lingkup yang memungkinkan untuk mempertimbangkan secara seksama dan penentuan apa yang tepat dan prioritas tertingginya. Sekali lagi ada berbagai pendekatan untuk menciptakan sebuah EA. *TOGAF* seperti pendekatan lain menyediakan metodologi untuk mengembangkan EA. *Development Method Architecture* (ADM) berguna dalam pembahasan pentingnya ruang lingkup dan bagaimana seseorang bisa memadukan model yang berbeda. Ada empat dimensi utama dalam ruang lingkup yang dapat didefinisikan dan terbatas (Stansfield, Orobato, Lubinski, Uggowitz, & Mwanyika):

1. Ruang lingkup atau fokus *Enterprise*: seberapa besar perusahaan tersebut serta berapa banyak perusahaan yang seharusnya difokuskan arsitekturnya?
2. Domain Arsitektur: deskripsi arsitektur lengkap harus berisi empat domain arsitektur (bisnis, data, aplikasi dan infrastruktur), tetapi keterbatasan sumber daya akan berdampak pada tingkat arsitektur.
3. Ruang lingkup vertikal atau level detail: berapa banyak arsitektur

yang memadai sebelum pelaksanaan kegiatan termasuk desain sistem, rekayasa sistem dan pengembangan sistem dapat dimulai?

4. Horizon waktu: horison waktu seperti apa untuk membangun arsitektur lengkap dan apakah hambatan sumber daya memerlukan pendefinisian Arsitektur Sasaran menengah yang memungkinkan kegiatan pelaksanaan untuk dimulai?

4.2. Arsitektur aplikasi terdistribusi

Ketika penggunaan Web untuk aplikasi bisnis mulai berkembang pada tahun 1995-an, arsitektur aplikasi terdistribusi (kemudian kadang-kadang disebut sebagai arsitektur *N-tier*) muncul. Hal ini tampak berbeda dari arsitektur client-server tradisional dimana database umumnya ditempatkan pada server yang terpusat dan thick client berisi user interface dan logika aplikasi yang diinstal di desktop setiap pengguna. Ada berbagai kelemahan dalam model ini, termasuk masalah skalabilitas karena koneksi database khusus dari setiap pengguna serta overhead operasional instalasi, distribusi dan pemeliharaan perangkat lunak klien pada desktop pengguna.

Arsitektur aplikasi terdistribusi benar-benar dimungkinkan oleh dua hal. Pertama, Web browser yang muncul sebagai tools di setiap desktop. Kedua, kelas perangkat lunak yang dikenal sebagai aplikasi server terungkap bahwa memungkinkan fungsionalitas aplikasi akan dikirimkan ke desktop melalui Web browser. Hilanglah biaya distribusi dan pemeliharaan bagi klien (web browser) baik terinstal dalam sistem operasi, atau paling tidak tersedia bebas untuk didownload dan diinstal dengan mudah. Masalah skalabilitas itu hilang karena lingkungan aplikasi server terpisah dalam berbagai komponen yang menangani permintaan pengguna sehingga mereka dapat ditingkatkan sesuai kebutuhan.



Gambar 2 menunjukkan berbagai tingkatan arsitektur ini.

- **User Tier** : biasanya terdiri dari browser Web yang sudah ada pada desktop atau dapat dengan mudah didownload dan diinstal secara gratis. Alternatif untuk browser Web dapat mencakup aplikasi thin client. Ini biasanya hanya menangani input dan output dari server (di mana sebagian besar pengolahan dan logika bisnis berada), dan seringkali dapat didownload bila diperlukan.
- **Application Tier** : biasanya termasuk aplikasi server yang menangani permintaan HTTP / HTTPS dari pengguna dan memproses permintaan tersebut dengan logika bisnis dan akses database sesuai kebutuhan melalui *data engine tier*. *Application Tier* umumnya diperuntukkan bagi layanan aplikasi seperti pengelompokan server aplikasi dan *load balancing* diantara mereka untuk skalabilitas dan ketahanan, sesi pengelolaan data pengguna, pengelolaan koneksi database bersama dan pengelolaan objek aplikasi berisi logika bisnis.
- **Data Engine Tier** : menangani pengolahan semua permintaan data, mulai dari menyimpan data terstruktur yang dikelola SQL atau menyimpan data terstruktur dan tidak terstruktur seperti sistem file. *Tier* ini umumnya diperuntukkan bagi pengelompokan untuk skalabilitas

dan ketahanan. Selain menyediakan lapisan abstraksi, memisahkan logika bisnis dari data, ada alasan lain untuk memisahkan secara fisik tier ini dari *application tier*. Hal ini termasuk kebutuhan hardware yang berbeda (yaitu, akses harddisk yang lebih dioptimalkan untuk server database dibandingkan RAM tambahan yang diperlukan oleh server aplikasi) dan menambahkan keamanan (pemisahan tier ini untuk segmen jaringan internal agar mengurangi risiko akses data penting ke luar yang tidak sah).

- **Data Storage Tier** : digunakan untuk penyimpanan data secara fisik. Dalam beberapa kasus berupa server fisik yang sama sebagai *Data Storage Tier*. Dalam hal lain, sebuah *Storage Area Network* (SAN) dapat digunakan untuk optimasi penyimpanan dan ketahanan.

4.3. Arsitektur Sistem Informasi RS

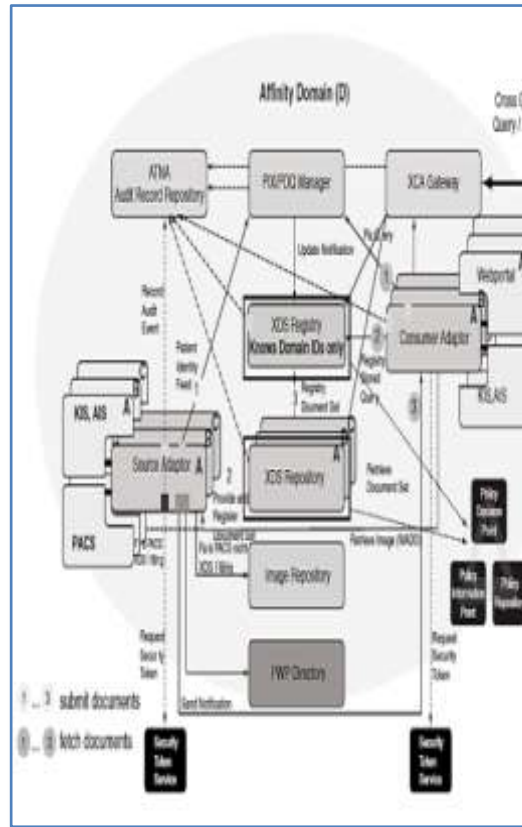
Umumnya ada lapisan atau domain dari arsitektur yang merupakan subset dari suatu arsitektur enterprise secara keseluruhan. Empat lapisan biasanya didefinisikan dalam model umum dari suatu arsitektur enterprise. Bagi developer dan mitra pelaksana untuk memiliki panduan yang cukup tentang empat domain tersebut menyediakan rincian yang berbeda.

1. *Organizational Architecture*
2. *Data Architecture*
3. *Applications Architecture*
4. *Technical Architecture*

Menurut definisi sebelumnya SIRS adalah subsistem sosio-teknis dari sebuah rumah sakit yang terdiri dari semua pengolahan informasi serta aktor manusia atau teknis yang terkait dalam peran masing-masing memproses informasi. Arsitektur dan perangkat pengolahan informasi pada umumnya terbagi kepada lapisan logis dan lapisan fisik dari perangkat SIRS. Pada setiap lapisan menjelaskan cara perangkat ini dapat dirakit dan diintegrasikan untuk

memberikan dukungan yang lebih baik bagi pengguna. Lapisan lapisan tersebut adalah:

- a. *Domain Layer: Data yang Harus Diolah Rumah Sakit*. Karena SIRS adalah subsistem dari rumah sakit yang berhubungan dengan pengolahan informasi atau lebih tepatnya pengolahan data.
- b. *Domain Layer: Hospital Functions*. Fungsi-fungsi rumah sakit juga dapat dianggap sebagai wakil proses. Fungsi rumah sakit dan jenis entitas ditafsirkan dan diperbarui secara lebih rinci.
- c. *Logical Tool Layer: Application Components Logical Tool Layer* pada lapisan SIRS, adalah komponen aplikasi yang mendukung fungsi RS.
- d. *Logical Tool Layer: Integration of Application Components*. SIRS biasanya terdiri dari komponen dari berbagai aplikasi yang perlu diintegrasikan untuk mencapai pengolahan informasi dengan kualitas tinggi.
- e. *Physical Tool Layer: Physical Data-Processing Systems*. Sistem pengolahan data secara fisik dapat berupa aktor manusia, non-komputer berbasis alat fisik, atau sistem komputer.
- f. *Physical Tool Layer: Integration of Physical Data-Processing Systems*. Jika pertukaran data satu sama lain dan berinteraksi, telah terinstal pada sistem pemrosesan data fisik yang berbeda pada lapisan tool fisik, data-data fisik sistem pengolahan harus berkomunikasi juga, oleh karena itu diperlukan integrasi.



Gambar 3. Gambaran Arsitektur berdasarkan pengertian beberapa profil Integrasi IHE. Aktor digambarkan sebagai kotak dan transaksi sebagai garis (Winter).

4.4 Arsitektur SaaS

Model SaaS meliputi konsep-konsep dalam arsitektur aplikasi terdistribusi tetapi lebih lanjut memperluas arsitektur untuk menyertakan komponen untuk memfasilitasi dan meningkatkan model bisnis. Sebuah vendor perangkat lunak tradisional terutama berkaitan dengan kemampuan aplikasi dan pelanggan mereka yang bertanggung jawab untuk mengoperasikan dan mengelola lingkungan yang bersangkutan di mana mereka menjalankan perangkat lunak.

Sebuah vendor SaaS, di sisi lain, adalah sama-sama memperhatikan pengoperasian dan pengelolaan lingkungan yang mendukung seluruh pelanggan mereka.

HIS Enterprise Architecture [5]		
Architecture Domain	Deliverables	Pertanyaan representatif yang ditujukan
1. Organizational Architecture	<ul style="list-style-type: none"> Business domains Business functions Business processes Governance, Policy, Resources 	<ul style="list-style-type: none"> Sapa pengambil keputusan utama, apa peran dan perilaku pengambilan keputusan tersebut? Apa saja pertanyaan penting sebagai pengguna yang harus mampu menjawab untuk membuat keputusan strategi dan keputusan harian? Apa inti proses bisnis, jasa pelayanan kesehatan, laboratorium, farmasi, yang diperlukan untuk mendukung pengambilan keputusan? Apa kebijakan dan hukum yang diperlukan untuk mendukung pengembangan awal dan implementasi SIRS? Sumber daya apa saja yang diperlukan untuk membangun kapasitas minimum agar SIRS berkelanjutan? Sapa yang akan bertanggung jawab untuk pemeliharaan integritas SIRS?
2. Data Architecture	<ul style="list-style-type: none"> Model Data Kamus Metadata Klasifikasi standar dan sistem 	<ul style="list-style-type: none"> Apa data umum dan inti yang diperlukan untuk mendukung informasi bagi pengambilan keputusan? Sumber data apa yang dapat dihubungkan untuk digunakan dari sistem operasional yang ada? Apa hubungan antara kelompok data esensial minimum dan program M.I. Egoital?
3. Applications Architecture	<ul style="list-style-type: none"> Software aplikasi Antarmuka antar aplikasi Antarmuka pengguna 	<ul style="list-style-type: none"> Aplikasi SIRS awal apa, yang harus dapat diberikan? Apakah instrumen pengumpulan data standar, layanan komunikasi data, analisis data dan pemodelan, atau generator laporan? Aplikasi apa yang telah dalam desain platform tunggal diintegrasikan dengan aplikasi yang dipertahankan sebagai sistem operasional yang terpisah? Bagaimana perantara yang dimiliki aplikasi untuk bisa dihubungkan? Bagaimana retabilitas user interface diintegrasikan?
4. Technical Architecture	<ul style="list-style-type: none"> Platform perangkat keras Jaringan Komputer Sistem Operasi Interoperabilitas 	<ul style="list-style-type: none"> Apa saja persyaratan untuk informasi yang akan ditransmisikan, data yang dimasukkan, ditandai, dikomunikasikan, dikelola dan didebetalisasi? Apa yang dimaksud dengan informasi minimum dan kapasitas teknologi komunikasi yang dibutuhkan untuk mendukung penyebaran informasi dan akses pada aplikasi? Bagaimana perangkat elektronik baru, jaringan komunikasi dan ICT terkait dapat dimanfaatkan selama 5 sampai 7 tahun ke depan?

Cloud Software as a Service (SaaS). Kemampuan yang diberikan kepada konsumen adalah dengan menggunakan aplikasi penyedia yang berjalan pada infrastruktur awan. Aplikasi dapat diakses dari berbagai perangkat klien melalui antarmuka *thin-client* seperti Web browser (misalnya, Web-based e-mail). Konsumen tidak mengelola atau mengendalikan infrastruktur awan termasuk jaringan, server, sistem operasi, penyimpanan, atau kemampuan aplikasi bahkan individu, dengan pengecualian terbatas khusus bagi pengguna untuk pengaturan konfigurasi aplikasi.

Gambar 4 menunjukkan diagram konseptual dari arsitektur yang lebih komprehensif. Sebuah tier distribusi ditambahkan untuk mengakui bahwa permintaan layanan mungkin perlu diarahkan pada lebih dari satu lingkungan operasi fisik karena berbagai alasan. Tier Komponen aplikasi

tambahan yang memperhitungkan fungsi tambahan penawaran SaaS yang matang yang mungkin diperlukantelah diidentifikasi. Masalah-masalah administrasi dan pemantauan disajikan sebagai komponen yang beroperasi di seluruh tingkatan. Komponen-komponen ini sangat penting untuk operasi yang efisien dari bisnis SaaS.



Gambar 4. SaaS Architecture

Tier distribusi ditambahkan ke model arsitektur SaaS untuk menangani *load balancing*. Istilah ini umumnya berhubungan dengan teknik untuk menyebarkan pekerjaan untuk mengoptimalkan pemanfaatan sumber daya, *throughput* atau waktu respon. Dengan bisnis SaaS ini tentu tepat untuk mempertimbangkan sehingga pelanggan anda dapat diberikan dengan tingkat pelayanan sebaik mungkin. Namun, ada alasan lain mengapa permintaan layanan mungkin perlu diarahkan ke berbagai lingkungan operasi fisik.

Salah satu alasannya adalah untuk memanfaatkan beberapa pusat data yang secara geografis berbeda untuk alasan kehandalan. Idenya adalah bahwa pelanggan yang berbeda dapat dialihkan ke situs utama yang berbeda. Semua situs yang digunakan untuk permintaan layanan produksi dan mereka bertindak sebagai situs alternatif antara satu sama lain, sehingga lebih efisien dalam memanfaatkan sumber daya dan meningkatkan ketersediaan pusat data. Alasan lain untuk routing permintaan layanan pelanggan tertentu dengan

lingkungan tertentu. Faktanya adalah bahwa data pribadi merupakan masalah besar dan berbagai negara telah mengambil pendekatan yang sangat berbeda untuk melindungi data warga mereka. Meskipun upaya, seperti program sertifikasi telah dikembangkan untuk memfasilitasi kepatuhan terhadap privasihukum, sejumlah pelanggan dari negara tertentu mungkin mengharuskan data perusahaan mereka secara fisik disimpan di negara mereka. Jika vendor SaaS ingin mengembangkan bisnis di negara tersebut, lingkungan produksi internasional yang berbeda mungkin diperlukan dan routing permintaan akan diperlukan.

5. KESIMPULAN

Palayanan kesehatan bagi Rumah Sakit dari hari ke hari harus terus ditingkatkan agar mampu meningkatkan pelayanannya kepada pasien serta mampu '*bersaing*' dengan Rumah Sakit lainnya. Pengolahan data di Rumah Sakit adalah pekerjaan yang berskala besar dan rumit serta melibatkan berbagai fungsi. Oleh karena itu RS memerlukan pemanfaatan Sistem Informasi untuk mengelola seluruh datanya dan menyajikannya sebaik mungkin bagi seluruh *stake-holder* Rumah Sakit.

Menerapkan sistem informasi tidak semudah yang dibayangkan para penggunanya, begitu juga pengguna di RS. Berbagai keterbatasan seperti sumber daya manusia, perangkat keras, perangkat lunak maupun teknologi serta alokasi biaya yang terbatas menjadi pemicu gagalnya penerapan sistem informasi di RS.

Pendekatan *Cloud Computing* memberikan peluang untuk menerapkan SIRS dalam berbagai keterbatasan sumber daya RS. Pendekatan SaaS untuk membangun solusi perangkat lunak sedang muncul dan berkembang. Sebagai teknologi baru, standarisasi terus dikembangkan, model ini pasti akan terus tumbuh dan berkembang. Melalui penerapan SIRS menggunakan SaaS akan mengurangi biaya investasi

baik hardware maupun software serta menyelesaikan masalah keterbatasan sumber daya manusia dalam bidang teknologi informasi.

Arsitektur SIRS akan memperhatikan;

- Organizational Architecture,
- Data Architecture,
- Applications Architecture dan
- Technical Architecture.

Sedangkan Arsitektur SaaS akan memperhatikan;

- User Tier,
- Application Tier,
- Data Engine Tier,
- Data Storage Tier, ditambah
- Distribution Tier.

Oleh karena itu Arsitektur SaaS pada layanan RS akan memberikan kontribusi pada *Data Architecture*, *Application Architecture* dan *Technical Architecture* bahkan berkontribusi juga pada *Organizational Architecture*. Tentu saja masih diperlukan kajian lebih lanjut tentang berbagai aspek seperti risiko keamanan data, data dan fungsi aplikasi apa saja yang akan dikelola dengan SaaS, aspek hukum, aspek kerjasama dengan penyedia layanan, serta strategi seperti apa yang harus diterapkan agar pemanfaatan SaaS dapat benar benar optimal.

REFERENSI

- Mell, P., & Grance, T. (2009, July 10). The NIST Definition of Cloud Computing Version 15.
- Oracle Technology Network. (n.d.). Retrieved April 12, 2012, from <http://www.oracle.com/technetwork/topics/whatsnew/index.html>
- Progress Software Corporation, White Paper: SaaS architecture. <http://www.progress.com>.
- Stansfield, S., Orobato, N., Lubinski, D., Uggowitzer, S., & Mwanyika, H. *The Case for a National Health Information System Architecture; a Missing Link to Guiding National Development and Implementation*.
- Winkler, V. (. (n.d.). *Securing the Cloud: Cloud Computer Security Techniques and Tactics*. Syngress is an imprint of Elsevier .
- Winter, A. a. *Health Information Systems - Architectures and Strategies Second Edition*. Springer.